

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 1.**

Fișierul *date.in* conține pe prima linie o valoare naturală nenulă  $n$ , iar pe următoarele  $n$  linii, separate printr-un spațiu, câte o pereche de numere naturale nenule reprezentând numărătorul, respectiv, numitorul unei fracții ( $2 \leq n \leq 10$ ). Să se determine suma acestor fracții. Numărătorul și numitorul fracției obținute în forma ireductibilă (valori mai mici sau egale decât 1000000000) vor fi scrise în fișierul *date.out* pe prima linie a acestuia separate printr-un spațiu.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	4	43 20	$\frac{7}{6} + \frac{1}{3} + \frac{1}{4} + \frac{2}{5} = \frac{43}{20}$
	7 6		
	1 3		
	1 4		
	2 5		

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 2.**

Fișierul *date.in* conține pe unica sa linie un număr natural  $n$  ( $n \leq 1000000000$ ). Să se determine „cifra de control” a acestui număr, efectuând suma cifrelor sale, apoi suma cifrelor acestei sume, etc., până se obține o sumă formată dintr-o singură cifră. Rezultatul obținut va fi scris în fișierul *date.out*.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	1971	9	$1971 \rightarrow 18 \rightarrow 9$

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 3.**

Fișierul *date.in* conține pe unica sa linie, separate prin spații, trei numere naturale nenule,  $a$ ,  $b$  și  $n$ ,  $0 \leq a < b \leq 10000$ ,  $0 < n \leq 10000$ . Să se determine primele  $n$  zecimale ale fracției  $a/b$ . Cifrele determinate vor fi scrise în fișierul *date.out* pe o singură linie.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	5 23 5	21739	$\frac{5}{23} = 0.21739\dots$

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 4.**

Să se determine toate numerele naturale prime de trei cifre care citite invers sunt tot numere prime. Rezultatele obținute vor fi scrise în fișierul *date.out*, câte unul pe fiecare linie a fișierului.

Ex.	<i>date.out</i>	Explicație
	101	două astfel de numere sunt 101 și 167
	...	
	167	
	...	

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 5.**

Un număr natural  $n$  este deosebit dacă există un număr natural  $m$  cu proprietatea că adunând acest număr  $m$  cu suma cifrelor numărului  $m$  se obține numărul  $n$ . Fișierul *date.in* conține pe unica sa linie un număr natural  $n$  ( $n \leq 1000000000$ ). Să se verifice dacă acest număr este deosebit. Rezultatul verificării va fi scris în fișierul *date.out* pe unica linie a acestui fișier sub forma unui mesaj corespunzător (DA sau NU după cum numărul are proprietatea respectivă sau nu).

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	1235	DA	1235 poate fi scris ca 1225+10
	<i>date.in</i>	<i>date.out</i>	
	20	NU	

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 6.**

Fișierul *date.in* conține pe unica sa linie un număr natural nenul  $n$  reprezentând numărul de pagini ale unei cărți ( $20 \leq n \leq 10000$ ). Să se determine câte cifre au fost folosite la paginarea cărții. Rezultatul obținut va fi scris în fișierul *date.out*.

Ex.	<i>date.in</i>	<i>date.out</i>
	100	192
	<i>date.in</i>	<i>date.out</i>
	15	21

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 7.**

Fișierul *date.in* conține pe unica sa linie, separate printr-un spațiu, două numere naturale nenule,  $a$  și  $b$  ( $1 \leq a, b \leq 1000000000$ ). Să se verifice dacă aceste numere pot fi termeni consecutivi ai șirului Fibonacci (1,1,2,3,5,8,...). Rezultatul verificării va fi scris în fișierul *date.out* pe unica linie a acestui fișier sub forma unui mesaj corespunzător (DA sau NU după cum cele două numere sunt termeni consecutivi ai șirului Fibonacci sau nu).

Ex.	<i>date.in</i>	<i>date.out</i>
	2 3	DA
	<i>date.in</i>	<i>date.out</i>
	89 55	DA
	<i>date.in</i>	<i>date.out</i>
	4 5	NU

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 8.**

Fișierul *date.in* conține pe unica sa linie un număr natural  $n$  ( $n \leq 1000000000$ ). Să se determine toate reprezentările posibile ale acestui număr ca sumă de numere naturale nenule consecutive. Aceste reprezentări vor fi scrise în fișierul *date.out*, câte una pe fiecare linie a fișierului; numerele din cadrul fiecărei reprezentări vor fi separate prin spații.

Ex.	<i>date.in</i>	<i>date.out</i>
	15	1 2 3 4 5 4 5 6 7 8
	<i>date.in</i>	<i>date.out</i>
	50	8 9 10 11 12 11 12 13 14

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 9.**

Un număr natural se numește „super prim” dacă numărul respectiv și toate prefixele acestuia sunt numere prime. Fișierul *date.in* conține pe unica sa linie un număr natural  $n$  ( $n \leq 1000000000$ ). Să se verifice dacă acest număr este „super prim” și, în caz afirmativ, să se determine prefixele sale. Rezultatul verificării va fi scris în fișierul *date.out* pe prima linie a acestui fișier sub forma unui mesaj corespunzător (DA sau NU după cum numărul este „super prim” sau nu); dacă numărul este „super prim”, fiecare următoare linie a fișierului va conține câte un prefix al numărului respectiv.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	239	DA 2 23	numărul 239 este prim; prefixele sale (2 și 23) sunt numere prime
	<i>date.in</i>	<i>date.out</i>	Explicație
	17	NU	numărul 17 este prim; prefixul său (1) nu este prim

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 10.**

Fișierul *date.in* conține pe unica sa linie un număr natural  $n$  ( $n \leq 1000000000$ ). Să se determine cel mai apropiat număr prim față de  $n$ . Programul va conține cel puțin un subprogram, iar rezultatul obținut va fi scris în fișierul *date.out*. Dacă sunt două numere prime egal departate de  $n$ , se poate afișa oricare dintre ele.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	1400	1399	1399 este numărul prim cel mai apropiat de 1400
	<i>date.in</i>	<i>date.out</i>	Explicație
	3019	3019	3019 este număr prim
	<i>date.in</i>	<i>date.out</i>	Explicație
	3000	3001	3001 și 2999 sunt numere prime egal departate de numărul 3000

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 11.**

Fișierul *date.in* conține pe prima sa linie un număr natural  $n$  ( $1 \leq n \leq 100$ ), iar pe următoarea linie, separate prin spații, cele  $n$  elemente ale unui vector de numere naturale. Să se determine câte din elementele vectorului dat sunt numere Fibonacci. Rezultatul obținut va fi scris în fișierul *date.out*.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	10	5	sunt cinci numere Fibonacci în vectorul dat;
	5 10 1 7 9 8 1 6 55 19		acestea sunt: 5, 1, 8, 1, 55

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 12.**

Fișierul *date.in* conține pe prima sa linie un număr natural  $n$  ( $1 \leq n \leq 100$ ), iar pe următoarea linie, separate prin spații, cele  $n$  elemente ale unui vector de numere naturale. Să se verifice dacă vectorul dat este o mulțime (în sensul cunoscut din matematică), dacă nu, să se transforme acest vector în mod corespunzător. Elementele vectorului rezultat se vor afișa pe prima linie a fișierului *date.out*, separate prin spații.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	10	1 2 9 4 6 5 20 3	elementele vectorului inițial nu sunt
	1 2 9 4 2 6 5 1 20 3		două câte două distincte
	<i>date.in</i>	<i>date.out</i>	Explicație
	7	1 2 9 4 6 5 3	elementele vectorului inițial sunt
	1 2 9 4 6 5 3		două câte două distincte

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 13.**

Fișierul *date.in* conține: pe prima sa linie un număr natural  $n$  ( $1 \leq n \leq 100$ ), pe a doua linie, separate prin spații, cele  $n$  elemente ale unui vector  $a$  de numere întregi, pe a treia linie un număr natural  $m$  ( $1 \leq m \leq 100$ ), iar pe a patra linie, separate prin spații, cele  $m$  elemente ale unui vector  $b$  de numere întregi. Să se afișeze pe prima linie a fișierului *date.out* câte din elementele vectorului  $b$  sunt strict mai mici decât toate elementele vectorului  $a$ .

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	10	3	în al doilea vector sunt trei elemente strict mai
	-3 2 9 4 2 6 -5 -1 20 3		mici decât toate elementele primului vector;
	8		acestea sunt: -6, -18, -6
	6 -5 0 -6 20 2 -18 -6		

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 14.**

În fișierul *date.in* se găsesc două numere mari (care pot avea mai mult de 10 cifre), câte unul pe o linie. Să se afișeze pe prima linie a fișierului *date.out* suma celor două numere.

Ex.	<i>date.in</i>	<i>date.out</i>
	45899200768797	45899251768664
	50999867	

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 15.**

Fișierul *date.in* conține pe prima sa linie un număr natural  $n$  ( $1 \leq n \leq 100$ ), iar pe următoarea linie, separate prin spații, cele  $n$  elemente ale unui vector de numere întregi. Știind că vectorul este format din două subsecvențe de numere ordonate crescător, să se ordoneze întregul vector prin interclasarea celor două subsecvențe. Elementele vectorului rezultat se vor afișa pe prima linie a fișierului *date.out*, separate prin câte un spațiu.

Ex.	<i>date.in</i>	<i>date.out</i>
	14	-12 -6 -5 -3 1 2 2 4 5 5 9 14 14 25
	-3 2 4 5 9 14 -12 -6 -5 1 2 5 14 25	

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 16.**

Fișierul *date.in* conține pe prima sa linie un număr natural  $n$  ( $1 \leq n \leq 100$ ), iar pe următoarea linie, separate prin spații, cele  $n$  elemente ale unui vector de numere întregi. Să se afișeze, separate prin spații, pe prima linie a fișierului *date.out* elementul (elementele) care apare (apar) de cele mai multe ori în vectorul dat.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	14	14 5	elementele 14 și 5 apar fiecare
	3 2 14 5 9 14 12 -6 5 1 12 5 14 2		de trei ori în vector

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 17.**

Fișierul *date.in* conține pe prima sa linie un număr natural  $n$  ( $1 \leq n \leq 100$ ), iar pe următoarea linie, separate prin spații, cele  $n$  elemente ale vectorului  $v$  de numere întregi. Să se construiască un vector  $w$ , astfel încât  $w[i]$ =numărul de apariții ale lui  $v[i]$  în vectorul  $v$ ,  $1 \leq i \leq n$ . Să se afișeze cei doi vectori, fiecare pe câte o linie a fișierului *date.out*. La afișare, elementele fiecărui vector vor fi separate prin câte un spațiu.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	10	3 2 14 5 9 14 2 -6 2 -6	elementul 3 apare o dată.,
	3 2 14 5 9 14 2 -6 2 -6	1 3 2 1 1 2 3 2 3 2	elementul 2 de trei ori etc.

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 18.**

Din fișierul *date.in* se citește  $n$  număr natural,  $1 \leq n \leq 20$ . Să se construiască o matrice cu  $n$  linii și  $n$  coloane ce se completează cu termenii șirului lui Fibonacci. Completarea se va face pe linii, de la stânga la dreapta. Nu se vor folosi structuri de date auxiliare. Matricea obținută se va scrie în fișierul *date.out*, fiecare linie a matricei pe câte o linie a fișierului, elementele afișate pe o linie separate prin spații.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	3	1    1    2	primii 9 termeni ai șirului lui Fibonacci sunt: 1, 1,
		3    5    8	2, 3, 5, 8, 13, 21 și 34
		13   21   34	

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 19.**

De pe prima linie a fișierului *date.in* se citește  $n$  număr natural,  $1 \leq n \leq 20$ , iar de pe următoarele  $n$  linii se citesc câte  $n$  numere întregi reprezentând elementele unei matricei. Se consideră că diagonalele matricei împart matricea în 4 zone: nord, sud, vest și est. Se cere să se calculeze suma elementelor impare din zona de nord a matricei (exclusiv elementele de pe diagonalele matricei) și să afișeze suma obținută în fișierul *date.out*.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	5	64	elementele impare din zona de nord a matricei
	45 2 63 8 10		sunt 63 și 1
	3 5 1 0 9		
	1 11 12 15 7		
	13 8 -1 41 19		
	-2 23 18 3 4		

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 20.**

Se citește din fișierul *date.in* un cuvânt de lungime cel mult 20 de caractere, format numai din litere mari. Să se afișeze în fișierul *date.out*, câte unul pe linie, toate cuvintele distincte ce se pot forma prin eliminarea câte unui singur caracter din cuvântul dat.

Ex	<i>date.in</i>	<i>date.out</i>	Explicație
	ACCES	CCES	nu contează ordinea în care se vor afișa cuvintele
		ACES	
		ACCS	
		ACCE	

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 21.**

Se citește din fișierul *date.in* un număr natural cu cel mult 9 cifre. Să se afișeze în fișierul *date.out* cel mai mare număr care se poate forma cu cifrele distincte ale numărului dat.

Ex	<i>date.in</i>	<i>date.out</i>	Explicație
	269363	9632	mulțimea cifrelor distincte este {2, 3, 6, 9}

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 22.**

Scrieți un program care citește de pe prima linie a fișierului *date.in* două numere naturale  $n, m$  cu  $n < m$  și scrie în fișierul *date.out* toate numerele prime din intervalul deschis  $(n, m)$ . Numerele se scriu în ordine crescătoare, câte 10 numere pe fiecare linie a fișierului, numerele dintr-o linie fiind despărțite între ele prin câte un spațiu. Dacă nu există niciun număr prim se va scrie un mesajul „Niciun număr prim” pe prima linie a fișierului.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	87 162	89 97 101 103 107 109 113 127 131 137 139 149 151 157	89 este primul număr prim mai mare decât 87, iar 157 este cel mai mare număr prim mai mic decât 162
	<i>date.in</i>	<i>date.out</i>	Explicație
	23 28	Niciun număr prim	În intervalul (23, 28) nu sunt numere prime

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 23.**

Se citește din fișierul *date.in* un număr natural  $n$  cu cel mult 9 cifre ( $n \geq 2$ ). Să se afișeze în fișierul *date.out* descompunerea numărului  $n$  în factori primi. Se va folosi cel puțin un subprogram în rezolvarea cerinței. Afișarea fiecărui factor și a puterii corespunzătoare lui se face pe câte un rând al fișierului, conform exemplului de mai jos.

Ex	<i>date.in</i>	<i>date.out</i>	Explicație
	48	factorul 2 la puterea 4 factorul 3 la puterea 1	$48 = 2^4 \cdot 3$

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 24.**

De pe prima linie a fișierului *date.in* se citește  $n$  număr natural,  $1 \leq n \leq 20$ . Să se construiască recursiv triunghiul de numere ca în exemplul de mai jos fără a folosi nicio instrucțiune repetitivă. Afișarea triunghiului se va face în fișierul *date.out*, conform exemplului.

Ex	<i>date.in</i>	<i>date.out</i>
	5	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5

**Atestat Informatică 2017 – Programare**  
**Biletul nr. 25.**

De pe prima linie a fișierului *date.in* se citește  $n$  număr natural,  $1 \leq n \leq 50$ , de pe a doua linie se citesc  $n$  numere reale distincte ordonate crescător reprezentând elementele tabloului unidimensional  $v$ , iar de pe a treia linie un număr real  $x$ . Să se afișeze în fișierul *date.out* pe ce poziție din tabloul  $v$  se găsește numărul  $x$ . Dacă valoarea  $x$  nu se găsește în tabloul  $v$  atunci se va afișa valoarea 0. Se va folosi o metodă eficientă de căutare.

Ex.	<i>date.in</i>	<i>date.out</i>	Explicație
	5	5	valoarea 45 se găsește pe poziția a 5-a în vectorul $v$
	12 14 20 24 45		
	45		
	<i>date.in</i>	<i>date.out</i>	Explicație
	4	0	valoarea 13 nu se găsește în vectorul $v$
	50 65 88 89		
	13		